

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号

特表2000-511649

(P2000-511649A)

(43) 公表日 平成12年9月5日 (2000.9.5)

(51) Int.Cl.⁷

G 0 9 C 1/00

識別記号

6 2 0

6 4 0

F I

G 0 9 C 1/00

データベース (参考)

6 2 0 A

6 4 0 B

審査請求 未請求 予備審査請求 有 (全 23 頁)

(21) 出願番号 特願平10-500255
 (86) (22) 出願日 平成8年6月5日 (1996.6.5)
 (85) 翻訳文提出日 平成10年12月7日 (1998.12.7)
 (86) 国際出願番号 PCT/FR96/00840
 (87) 国際公開番号 WO97/47110
 (87) 国際公開日 平成9年12月11日 (1997.12.11)
 (81) 指定国 EP(AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), CA, CN, JP, US

(71) 出願人 ジェムブリュス エス. セー. アー.
 フランス共和国, エフー13881 ジェムノ
 セデックス, ボアット ポスタル 100,
 アヴニユ デュ ピック ドゥ ベルター
 ニュ, パルク ダクティヴィテ ドゥ ジ
 ェムノ
 (72) 発明者 ムレイ, ダヴィッド
 フランス共和国, エフー75011 パリ, リ
 ュ パフロア, 30
 (72) 発明者 ナカッシュ, ダヴィッド
 フランス共和国, エフー75009 パリ, リ
 ュ シャブタル, 7
 (74) 代理人 弁理士 太田 恵一

(54) 【発明の名称】 公開鍵暗号方法

(57) 【要約】

本発明は、値 $G^k \pmod{p}$ を算出する離散対数に基づく公開鍵暗号方法に関するものである。本発明により、乗算回数を減らすために2つの方法が提案され、一方は、値1の少ないビット数であるがシステム全体としてのセキュリティを維持するには十分な長さの「中空の」指数 k を生成するものであり、他方は、特定の指数について同一の内積が繰り返されることがないように指数を互いに維持しつつ g 乗する並列計算を行うものである。本発明は、デジタル署名の生成、認証および暗号化に適用される。

【特許請求の範囲】

1. p はモジュールと呼ばれる素数であり、 k は n ビットの長さにたいてい等しい乱数であり、 g はベースと呼ばれる整数である $g^k \pmod{p}$ を算出し、エンティティ E が、かかる値を利用して別のエンティティとの間での署名を交換することを含む認証および／または署名および／または暗号化オペレーションを実現する離散対数に基づく公開鍵の暗号方法であって、

— N ビット長 (N は $n + b$ ビットに等しい) の乱数指数値 k を生成するステップと、

— 前記指数のハミング重み C を計算し、この値を予め固定値 h と比較するステップと、

— 乱数値 k が条件: $C \geq h$ を満たすか否かを確認するステップと、

— このハミング重みが h より小さい場合に乱数値 k を拒否し、この条件を満たす指数が得られるように新たな乱数を生成するステップと、

— 一方、逆の場合にはこの値を保持するステップと、

— 保持した値によって式 $g^k \pmod{p}$ を算出するステップと、

— 他のエンティティとの間で電子署名を取り交わす際にこの式を用いるステップと、エンティティについて含んでいることを特徴とする公開鍵暗号方法。

2. 満たすべき条件が $c = h$ であることを特徴とする請求項 1 に記載の方法。

3. p はモジュールと呼ばれる素数であり、 k は n ビットの長さに等しい乱数であり、 g はベースと呼ばれる整数である $g^k \pmod{p}$ を算出し、エンティティ E が、かかる値を利用して別のエンティティとの間での署名を交換することを含む認証および／または署名および／または暗号化オペレーションを実現する離散対数に基づく公開鍵の暗号方法であって、

— 以下の式

$$k_j = \sum a_i 2^i$$

で表される重み a_i の n ビットの乱数指数値 k_j の全体を生成するステップと、

— ある指数について算出された g の内積が中で機能する他の指数に対して用いられるように指数と組み合わせて、 g^{2^i} の内積を並列計算するステップと、

ー与えられた各 k_j について、まだ計算されていない内積 g を算出し、これらの内積を再度グループ分けして所望の式 $g_{k_j} \pmod{p}$ を得るステップと、
 ー他のエンティティとの間で電子署名を取り交わす際にかかる式を用いるステップと、
 を含むことを特徴とする公開鍵暗号方法。

4. 並列計算および再グループ分けするステップが、以下のオペレーションすなわち、

ー2ごとに指数を組み合わせ、それらの共通の部分を反映する指数 k_c を得て、結果として得られた組み合わせを再度繰り返し、

ー各 k_c 値について、

$$G_{k_c} = g^{k_c} \pmod{p}$$

になるように値 G_{k_c} を計算、

ー指数 k_j と、前記指数が属する前記組み合わせによって得られた指数 k_c とを組み合わせ、共通の部分をなくして異なる部分のみを保持し、

ー与えられた指数 k_j と与えられた指数 k_c との間の異なる部分を反映している指数 k_{kj} を定義し、

値 $G_{k_{kj}}$ を

$$G_{k_{kj}} = g \pmod{p}$$

になるように算出し、

ー各繰り返し時に得られた値 G_{k_c} の間の乗算によって式 $G_{k_j} \pmod{p}$ を求めることを含むことを特徴とする請求項3に記載の方法。

5. 並列計算および再グループ分けするステップが以下のオペレーションすなわち、

ー指数同士を組み合わせ、共通の部分を有する指数の考えられる組み合わせのあらゆる部分集合を形成し、

ー組み合わせの各部分集合について、重みのヌルではないビットが、考えられる組み合わせ

わせの同じ重みのヌルではないビットに対応している共通の部分を反映している

指数 k_c を定義し、

—各 k_c 値について $G_{k_c} = g^{k_c} \pmod{p}$ になるように値 G_{k_c} を算出し、

—各指数 k_j と、組み合わせの部分集合について各々得られた全ての指数 k_c とを組み合わせ、指数 k_j が属するこれは共通の部分はなくして異なる部分のみを保持し、

—与えられた指数 k_j と与えられた指数 k_c 間の異なる部分を反映している指数 k'_j を定義し、

— $G_{k'_j} = g^{k'_j} \pmod{p}$ になるように値 $G_{k'_j}$ を算出し、

—各 k_j について値 $G_{k'_j}$ と G_{k_c} との間で乗算を行うことによって式 $g^{k_j} \pmod{p}$ を求めることを含むことを特徴とする請求項3に記載の方法。

【発明の詳細な説明】

公開鍵暗号方法

本発明は、 p 法値の計算を行う離散対数に基づいたいわゆる公開鍵のといわれる暗号方法を目的とするものである。

本発明は、メッセージのデジタルサインの生成、2つの単位（実体）間での認証あるいはデータの暗号化に用途がある。

このような方法において安全性は、特定の関係、とりわけ離散対数を逆にするという、そこに認められる極端な困難性に基づいている。

この問題点は、以下 $y = g^x \bmod p$ （ここで、 y は g^x を p で除した余りであると定義できる）と記述する数学的関係 $y = g^x \bmod p$ から考えて、 p 、 g および y が既知である場合に、 x を再認識することにある。この問題は、 p のサイズが 512 ビットであるかまたはそれを越えると、かつ x のサイズが 128 ビットであるかまたはそれを越えると、現在の知識では解決することは不可能である。

このようなシステムでは、一般に、モジュールを構成している大きなサイズの番号 p を提供する認証機関が存在する。この認証機関は、 g によって得られる全体、すなわち数字 $g^x \bmod p$ の形成された全体のような、ベースとよばれる整数 g も選択するが、この時 x は間隔 $[0, p-1]$ に属す、すなわち少なくとも 2^{128} の最大サイズの部分集合である。

パラメータ p および g はいわゆる「公開」であって、すなわち、これらは認証機関のユーザー全てに対して認証機関によって提供されるものである。

特定の変形例によれば、これらのパラメータは各ユーザーによって個別に選択され、この場合、公開鍵の完全な一部をなすものである。

暗号システムを用いる場合における主な欠点は、実行されている複雑な計算のために、比較的多くの計算および記憶手段が必要となるということにある。

事実、値 $g^k \bmod p$ の計算とは、乗算モジュールを実現することであるが、これは計

算時間およびメモリ容量においても費用がかかる。標準的なマイクロプロセッサしかを用いていない簡単な電子装置では、この種のオペレーションはほとんど実

現できない。

この種の計算専用のプロセッサを有する電子装置に関しては、それでもなお、中間値を得るのに必要な計算時間およびメモリ容量には制限するのが望ましい。

事実、値 $g^k \bmod p$ の計算は、一般に、英語の略語 SQM（平方累乗）として知られる従来の「累乗」方法によって比較的費用のかかるものであるが、なぜならそれが平均 $3/2 \log_2(p)$ 乗算と等しい為である。

この方法によれば、 k が n ビット長である時 g のすべてのべき、すなわちすべての平方： $g^0, g^1, g^2, \dots, g^n$ を計算し、次に、これらのべきの間で必要な乗算を行う。（例えば、 $g^{17} = g^1 \cdot g^{16}$ ）。

単純な「平方累乗」方法によれば、単純な g^k にも $n/2$ 乗算および n 平方が必要である。

N 個の署名を一度に送信しなければならない場合には、 $N g^k$ を生成し、したがって並列計算を行う。

並列「平方累乗」方法では、 $N \times n/2$ 乗算および n 平方が必要である。

E. BRICKELL らによって提案された、略語で BGKW と呼ばれる方法では、累乗方法を用いる場合に乗算回数を減らすことができるが、予め算出された多数の定数の記憶装置が必要があり、したがって、極めて不利益をもたらすような記憶装置のメモリー容量を備える必要がある。

この方法に N 値の並列計算を導入することは、中間値を保持するために多数のレジスタを用いることを示している。

したがって、この方法は、極めて短い時間に多数の署名を生成することが問題となる状況にある場合には、さらに制約を生じさせるものになる。なぜならこの場合、並列計算が導入されるからである。

本発明の目的は、これらのすべての欠点を改善することにある。本発明は、すべての暗号システム、特にマイクロプロセッサのチップカードタイプの携帯式装置による、暗

号的アルゴリズムの実行のために、計算時間とメモリ容量について柔軟で費用のかからない解決策をもたらすものである。

本発明の第1の目的によれば、提案されている暗号方法は、利用されている暗号スキーム（シュノールまたはE1ギャマル）に従えば計算時間において15～40%あるいはそれ以上のゲインが得られるように、乗算モジュールの回数を減らすことを可能とする。

本発明によれば、乗算回数を減らすために2つの解決策が提案される。一つは、わずかな1ビットとともにであるが、システムのセキュリティを保護するのに十分な「中空の」指数 k を生成することであり、他方は与えられた指数について「べき」の計算を行わないように、指数同士を組み合わせることによってべき g の並列計算を行うことである。

本発明は、特に、値 $g^{k \bmod p}$ の計算を導く離散対数に基づく公開鍵暗号方法を目的としている。ここで、 p はモジュールと呼ばれる素数であり、 k は通常 n ビット長のランダム値であり、 g はベースと呼ばれる整数であり、エンティティ E が認証および／または署名および／または暗号化のオペレーションを実現し、この値が介入する別のエンティティとの間での署名の交換を含むものであり、かかる暗号方法は、

- N が $n+b$ ビットに等しい N ビット長のランダム指数 k を生成するステップと、
- この指数のハミング重み C を計算し、この値を予め定められた値 h と比較するステップと、
- ランダム指数値 k が $C \geq h$ の条件を満たすか否かを確認するステップと、
- ハミング値が h より小さい場合にランダム値 k を除去し、この条件を満たすような指数が得られるまで新たな指数を生成することを再び開始するステップと、
- あるいは、逆の場合にはこの値を保持するステップと、
- 保持した値から式 $g^{k \bmod p}$ を算出するステップと、
- 他のエンティティとの間で電子署名を取り交わす際にかかる式を用いるステップと、

を含む。

また、本発明は、以下のような値 $g^{k \bmod p}$ の計算を導く離散対数に基づく公

開鍵暗号方法にも関するものである。ここで、 p はモジュールと呼ばれる素数であり、 k は通常 n ビット長のランダム値であり、 g はベースと呼ばれる整数であり、エンティティ E が認証および／または署名および／または暗号化のオペレーションを実現し、このタイプの複数の値が介入する別のエンティティとの間での署名の交換を含み、かかる暗号方法は、

以下の式

$$k_j = \sum a_i 2^i$$

で表される重み a_i の n ビットのランダム指数 k_j の全体を生成するステップと、
 ーある指数についてすでに算出されたべき g が、それが介入する別の指数に役立つように、指数を組み合わせて、 $g 2^i$ の「べき」を並列計算するステップと、
 ー与えられた各 k_j について、まだ計算されていないべき g を計算し、これらのすべての「べき」を再度グループ分けして所望の式 $g^{k_j} \bmod p$ を得るステップと、
 ー他のエンティティとの間で電子署名を取り交わす際にかかる式を用いるステップと、を含む。

本発明の一実施態様によれば、並列計算および再グループ分けのステップは以下のオペレーション、

ー2つごとに指数を組み合わせ、その共通部分を反映する指数 k_c を得て、得られた組み合わせの結果について組み合わせを繰り返すステップと、
 ー各 k_c 値について、

$$G_{k_c} = g^{k_c} \bmod p$$

になるように値 G_{k_c} を計算するステップと、

ー共通の部分をなくし、異なる部分のみを保持するようにこの指数が組み合わせによって得られた指数 k_c を指数 k_j と組み合わせるステップと、

ー与えられた指数 k_j と与えられた指数 k_c との間の異なる部分を反映している指数 $k'_{j,j}$ を定義し、

ー値 $G_{k'_{j,j}}$ を

$$G_{k'_{j,j}} = g^{k'_{j,j}} \bmod p$$

になるように算出し、

—各繰り返し時に得られた値 G_{k_c} の間で乗算を行うことにより式 $G_{k_j} \bmod p$ を求めるステップを含む。

本発明の第2の実施態様によれば、並列計算および再グループ分けのステップは以下のオペレーション、

—共通の部分をもつ指数の可能な組み合わせの部分集合を形成するように指数を組み合わせるステップと、

—与えられた重みのヌルではないビットが、考慮対象となる組み合わせの同じ重みのヌルではないビットに対応しているような組み合わせの各部分集合について共通部分を反映している指数 k_c を定義し、

—各 k_c 値について $G_{k_c} = g^{k_c} \bmod p$ になるように値 G_{k_c} を算出し、

—各指数 k_j と、これに対してこの指数 k_j が共通部分をなくして異なる部分のみを保持するように属する組み合わせの各部分集合について、各々得られた指数 k_c 全てを組み合わせるステップと、

—与えられた指数 k_j および与えられた指数 k_c の間の異なる部分を反映している指数 k'_j を定義し、

— $G_{k'_j} = g^{k'_j} \bmod p$ になるように値 $G_{k'_j}$ を算出し、

—各 k_j について値 $G_{k'_j}$ と G_{k_c} との間で乗算を行うことによって式 $G_{k_j} \bmod p$ を求めるステップを含む。

本発明の他の目的によれば、指数間での共通の部分を得られるようにする組み合わせは、論理和「AND」によって実現される。

本発明の他の目的によれば、異なる部分を得られるようにする組み合わせは、「排他

的論理和」論理関数によって達成される。

本発明の他の特徴および利点は、添付の図面を参照して説明する例示的かつ非限定的な実施例による説明を読むことで明らかになる。

—図1は、本発明を実施するのに適したシステムを示す基本図である。

—図2は、第1の実施例における方法の主なステップを示す機能図である。

—図3は、本発明の第1の実施態様による第2の実施例における方法の主なステップを示す機能図である。

—図4は、本発明の第2の実施態様による第2の実施例における方法の主なステップを示す機能図である。

図1に、本発明の目的である暗号方法の実施システムの基本図を示す。

このシステムは、少なくとも1つの他のエンティティE2と電子信号を交換したいと考えているエンティティE1からなる。各エンティティには、中央処理装置(CPU)11、30と、通信インタフェースと、揮発性メモリ(RAM)13、32および／または書き込み不可メモリ(ROM)14、34および／または書き込み可能なまたは再書き込み可能な不揮発性メモリ(EPROMまたはEEPROM)15、33と、アドレスバス、データバスおよび制御バス16、35を備えている。

処理装置および／またはROMは、本発明の目的たる方法において、すなわち、認証セッションの時、電子署名生成時、あるいは他のエンティティに送られる電子信号の暗号化時に行われる計算ステップの実施に対応するプログラムまたは計算リソースを有している。

処理装置またはROMは、モジュールの乗算、加算および減算の実施に必要なリソースを有している。

処理装置および／またはROMは、各暗号化アルゴリズム専用に使われる暗号機能とおよびパラメータgおよびpを含んでいる指数 k_j は、認証機関によって再書き込み可能メモリに予め記憶させることができ、あるいは、乱数発生器または秘密乱数ソース k_0 から順次生成することもできる。さらにエンティティE1は秘密鍵xを有している。

本発明は、口座上で行われるトランザクションにおいて厳格なセキュリティが必要と

される銀行分野で実施される暗号システムに特に利用される。また、他のエンティティから電子信号形式として送信されるメッセージの送信を認証したい場合にも同様である。さらに、他のエンティティとの信号の交換時にメッセージに署名

する必要がある場合も同様である。

実際、トランザクションを行いたいと思っているエンティティは、例えばチップカードなどの集積カードであり、この場合は目的地となるエンティティは銀行端末などである。

説明の後半は、本発明が離散アルゴリズムに基づく任意の暗号システムに適用できるという理解できるのでデジタルメッセージへの署名にこの方法を適用することについて述べるものである。

本発明による方法は、特にメモリ容量が少ない環境に適している、乗算回数を大幅に減らす第1の解決策を提案するものである。

この場合、原則は、指数のランダムな性質を当然維持したままで、選択されるハミング重みが可能な限り最も軽いという意味で「中空の」指数 k_j を生成するということである。

この目的のために、本発明による方法は、必要に応じて順次指数 k_j を生成するか、あるいは交換がなされる前に予め指数 k_j を生成することにある。この場合、当然これらの指数は記憶される。生成された指数は n ビットの長さではなく $n+b$ ビットを超える長さであり、以下において定義する条件を満たす。

$n+b$ の指数 k が生成されると、本方法は次にかかる指数のハミング重み C を算出し、続いてこれを予め定めた値 h と比較する。

比較結果が $C \geq h$ である場合には、指数は保持されエンティティによって利用されるが、このエンティティは、式 $g^k \bmod p$ を算出し、式を例えば署名として利用する様なデジタル信号を送信する際にこの式を利用する。

パラメータ C が必要な条件を満たさない場合には、対応する指数 k は除外され、新たな指数が生成され、この条件を満たす指数 k が得られるまで、条件についての確認ス

テップが再度実行される。

このように、この解決方法では、より小さな指数を用いた場合と全く同じセキュリティレベルを保持したままより少ない乗算を行うことを可能にするものである。

乗算回数を最大限に減らすことができる特別な実施例においては、 $c = h$ が選択される。

事実、ハミングの重みが h である $n+b$ ビット長（ $n = \log_2 p$ の場合）の指数の場合、つまりこの指数が n ビットの場合と同じ結合数を有する場合、次の関係が立証されなければならない。

$$2^n \leq c^h_{n+b} \text{ と、}$$

$$(N+b) / 2+h \leq n$$

（実行する計算数を削減することが出来ることを条件とする。）

$$\text{つまり、} 2n \leq (n+b)! / (n+b-h)! h!$$

および

$$b+2h \leq n$$

決定する b と h の数は与えられた1個の n （たとえば $n=160$ ）のこの二重条件付不等式を解いて求められる。

例証として、本発明による方法の結果と既知の方法とを比較してみた。

$n=160$ ビットのシュノールのアルゴリズムの場合と、 $n=512$ ビットのE1ギャマル アルゴリズムの場合。これらの結果を下表に示す。

変数	シュノール	E1ギャマル 計算時間	E1ギャマル メモリ空間
乗算	62 (h)	187 (h)	199 (h)
平方	87 (b=15)	279 (b=52)	273 (b=35)
力	149	469	472
ゲイン	6.8%	9.4%	7.8%

n ビットの指数によって覆われた署名空間にかかる応力を、得ることを希望するセキュリティのレベルにより α 関数によって減少させることが出来る。
よって、パラメータ n 、 h 、 b は条件(1)を満たす必要がある。

$$(1) 2^{n-a} \leq (n+b)! / (n+b-h)! h!$$

この場合、 $(n+b)$ ビット長の異なる確率変数から同一の署名を発生させる可能性を有する。

事実、 2^{80} は可能な様々なアタックを妨げるのに十分であり、したがって $n-\alpha$
 $=100$ は許容出来る値である。

変数 力	中空の指数 計算時間	中空の指数 メモリ空間	単純な平方累乗
乗算	37 (h)	49 (h)	$n/2$
平方	$n/2 + 7$ ($b=14$)	$n/2 + 2$ ($b=4$)	$n/2$
合計	$n/2 + 44$	$n/2 + 51$	n

この実行変数は、平方のコスト（計算時）がモジュール乗算のコストよりも少ないためますます注目に値する。

一般には次のようになる。

計算する平方の数 s が $s/2 \leq m \leq s$ で、 m が乗算の数である場合、この極端な二つケースは、 $m=s$ と $m=2s$ となる。

この二つの極端なケースを比較した結果を下表に表す。

変数 力	中空の指数 計算時間	中空の指数 メモリ空間	単純な平方累乗	ゲイン
シュノール ($m=2s$)	124	131	160	22.5%
EIギャマル ($m=2s$)	300	307	512	41%
シュノール ($m=s$)	204	211	240	15%
EIギャマル ($m=s$)	556	563	728	24%

シュノールとEIギャマルの図表にこの手順を適用した場合に求められたゲインは、単純に乗算した乗法と比較した場合も、1次数のコストが乗算コストと同じと

考えられる場合にも、極めて大きい。

別の実施方法によると、この手順はメモリ空間に関し、特に特別な応力のないシステムに適用される。

この実施方法においては、次数の計算を一回に限るために g の異なるべきの並列計算を行い、同一計算を数回にわたり実行しないように、指数を結合する。

本発明をよりわかりやすくするために、2つのべきの計算方式を説明する。

$k_j = \sum a_i 2^i$ この場合の k_j は乱数として求められた。(つまり、乱数発生器に基づき生成された) または、

$$k_k = k_j = \sum b_i 2^i$$

本手順に従って、 $k_c = \sum a_i b_i 2^i$ が k_j と k_k の間の共通部分を反映し、係数 a_i が 1 か 0 となるように、指数 k_c を規定するように指数 k_j と k_k を結合する。

指数 k_c は k_j と k_k の共有部分に該当する。つまり、

$$k_j = 1 \times 2^{10} + \dots + 0 + 1 \times 2^0 \text{ と、}$$

$$k_k = 1 \times 2^{10} + 0 + 0 \dots + 1 \times 2^0$$

$$k_c = 1 \times 2^{10} + 0 + \dots + 1 \times 2^0$$

本手順に従って、論理関数 ET によって指数 k を k_c とする。

次に、指数 k_j と指数 k_c 間の確実な部分を決定する2番目の組み合わせを行う。

同様に、指数 k_k と k_c 間の確実な部分も算出する。

排他的論理和として実行された組み合わせを $k_j \oplus k_c$ および $k_k \oplus k_c$ とする。

次の次数を並列計算する。

$$G_{kj} = g^{k_j \oplus k_c} \bmod p$$

$$G_{kk} = g^{k_k \oplus k_c} \bmod p$$

$$G_{kc} = g^{k_c} \bmod p$$

$g^{k_j} \bmod p$ と $G^{k_k} \bmod p$ を求めるには、次の演算を実行する。

$$1) G_{kj} \times G_{kc} \bmod p$$

$$2) G_{kk} \times G_{kc} \bmod p$$

2つの累乗を与えた上記の例と同様、 n 乗算ではなく平均して約 $3n/4$ の乗算を実行する。ゲインは25%となる。

本発明による手順は、最大数の指数の組み合わせで広げることが出来る。このような拡大は、図3と4に示すフローチャートで説明する2つの実行方法に従い導入することが出来る。

この場合、本発明は特に大量の署名を発生させる必要のある場合に特に適用される。

第一の実行方法により、下表により表される木構造のように2つずつ指数の組み合わせを行う。

$$\begin{array}{rcll}
 k_j & a_1 & a_2 & a_3 \quad a_4 \\
 k_c & b_1 = a_1 \cdot a_2 & & b_2 = a_3 \cdot a_4 \\
 & & c_1 = b_1 \cdot b_2 &
 \end{array}$$

この組み合わせにより、上記の例と同様、指数 k_c を k_j 間の共通部分に反映させることが出来る。

簡単に記述するためには、指数 k_j を a_1 、 a_2 、 a_3 、 a_4 とする。

指数 k_c をツリーの-1レベルで b_1 と b_2 とし、ツリーの-2レベルで c_1 とする。

$a_1 \cdot a_2$ 、 $a_3 \cdot a_4$ の組み合わせを論理関数ANDにより実行する。

このように構成されているツリーの各レベルで組み合わせを反復する。ビットの単純な統計分布によりツリー内に進むにつれ乗算回数が減少する。 $n/3$ 乗算により実行する計算の成果が減じられる。

上述のように、各レベルで G_{kc} の値が決定される。

この結果、次の式が得られる。

$$\begin{aligned}
G_{a1} &= g^{a1} \bmod p \\
G_{a2} &= g^{a2} \bmod p \\
G_{b1} &= g^{b1} \bmod p \\
G_{b2} &= g^{b2} \bmod p \text{ または } G_{b2} = G_{b1} \cdot G_{c1} \bmod p \\
G_{c1} &= g^{c1} \bmod p \\
G^{a1} \bmod p &= G_{a1} \times G_{b1} \bmod p = G_{a1} \times G_{c1} \bmod p
\end{aligned}$$

実際、 $g^{a1} \bmod p$ は $G_{a1} \times G_{b1} \bmod p$ の関により求められる。また $g^{a2} \bmod p$ は $G_{a2} \times G_{b1} \times G_{c1} \bmod p$ の関により求められる。

第二の方法によれば、指数を可能なあらゆる組み合わせの部分集合をなすように組み合わせるが、すなわち、例えば、指数 $kj:a,b,c$ では、 ab,ac,bc,abc の組み合わせを作る。

つまり論理関数ANDを a と b 、 a と c 、 b と c と a,b,c の間に行うことにより、これらの部分集合について共通部分を定められるような組み合わせをつくる。このようにして累乗指数 k_c を、各部分集合について定める。

すべての $G_{k_c} = g^{k_c} \bmod p$ 値の並列計算ができ、それらの K^c は初めの K に対してわずかに1ビットを有し、従って、それらについてのモジュラー変換も早い。

次に、一つの指数と、上述の組み合わせとの間で共通部分をとりのぞくことからなるまた別の組み合わせをつくる。

これらの組み合わせは論理関数で行う。このようにして次が得られる。

$$ka = a \text{ xor } abc \text{ xor } ac \text{ xor } ab$$

$$kb = b \text{ xor } abc \text{ xor } ab \text{ xor } bc$$

$$kc = c \text{ xor } abc \text{ xor } ac \text{ xor } bc$$

次に $G_{k_j} = g^{k_j} \bmod p$ を求めることができ、この k_j はさらにイニシャル k_c よりも1ビットが少なく、そのモジュラー変換速度はさらに早い。

最後に $g^{k_j} \bmod p$ の式が k_j によって得られる。

この第二の実施態様で得られた署名発生 N の場合、演算は、次のように示される。

n/N 二乗+ $n(2^N-1)/N2^N+(2^{N-1}-1)$ 乗算

次の表は、掛け算二乗、平行乗算二乗、および本発明のような周知の方法と比較した場合の結果を示している。

方法 効果時間	平方累乗	並列平方累乗	二分木 指数組み合わせ
二乗	$N(n-1)$	$n-1$	$n-1$
乗算	$N(n/2-1)$	$N(n/2-1)$	$Nn/3$
トータル	$N(3n/2-2)$	$N(n/2-1) + n-1$	$Nn/3+n-1$
$N \gg n$ についての効果	100%	33%	22%

署名Nの発生に適用される場合における与えられた第一の実施態様（木構造分類）はメモリー容量ではコストが安い。

指数4の二分木については計算に、 $\log_2(p)$ ビットの8つのレジスタが必要となるの

であろう。

与えられた第二の実施態様（N分類）が、計算時間において費用が非常に安いのは、乗算の回数において最適なためである。

3つの指数の場合、計算のために $\log_2(p)$ ビットの8つのレジスタが必要であろう。

【図1】

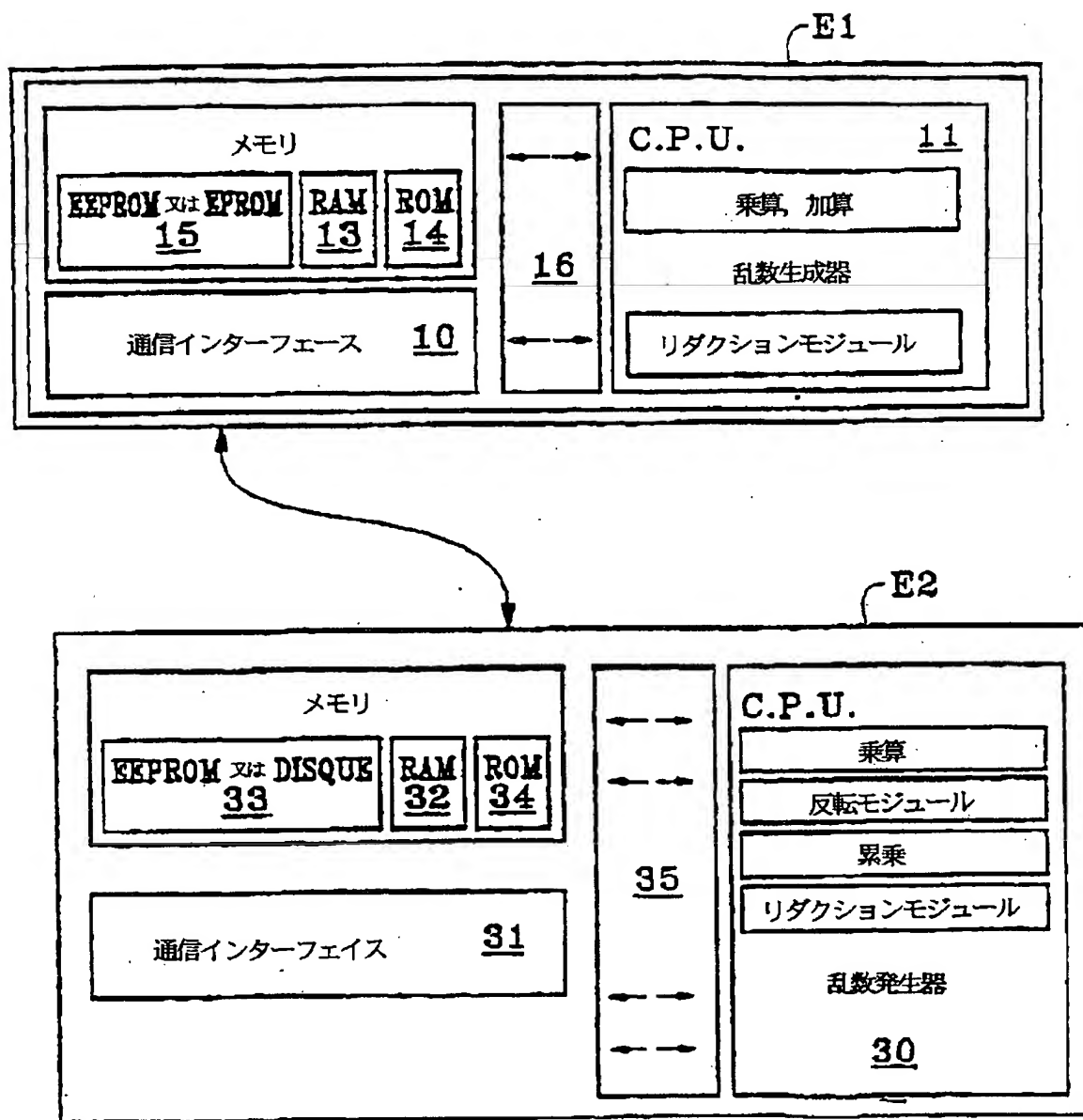


FIG.1

【図2】

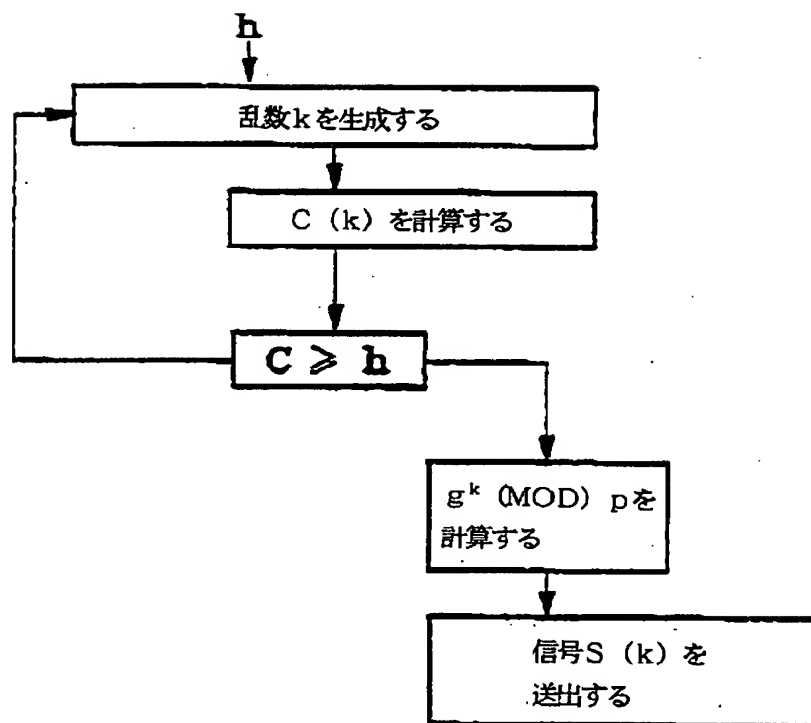


FIG.2

【図3】

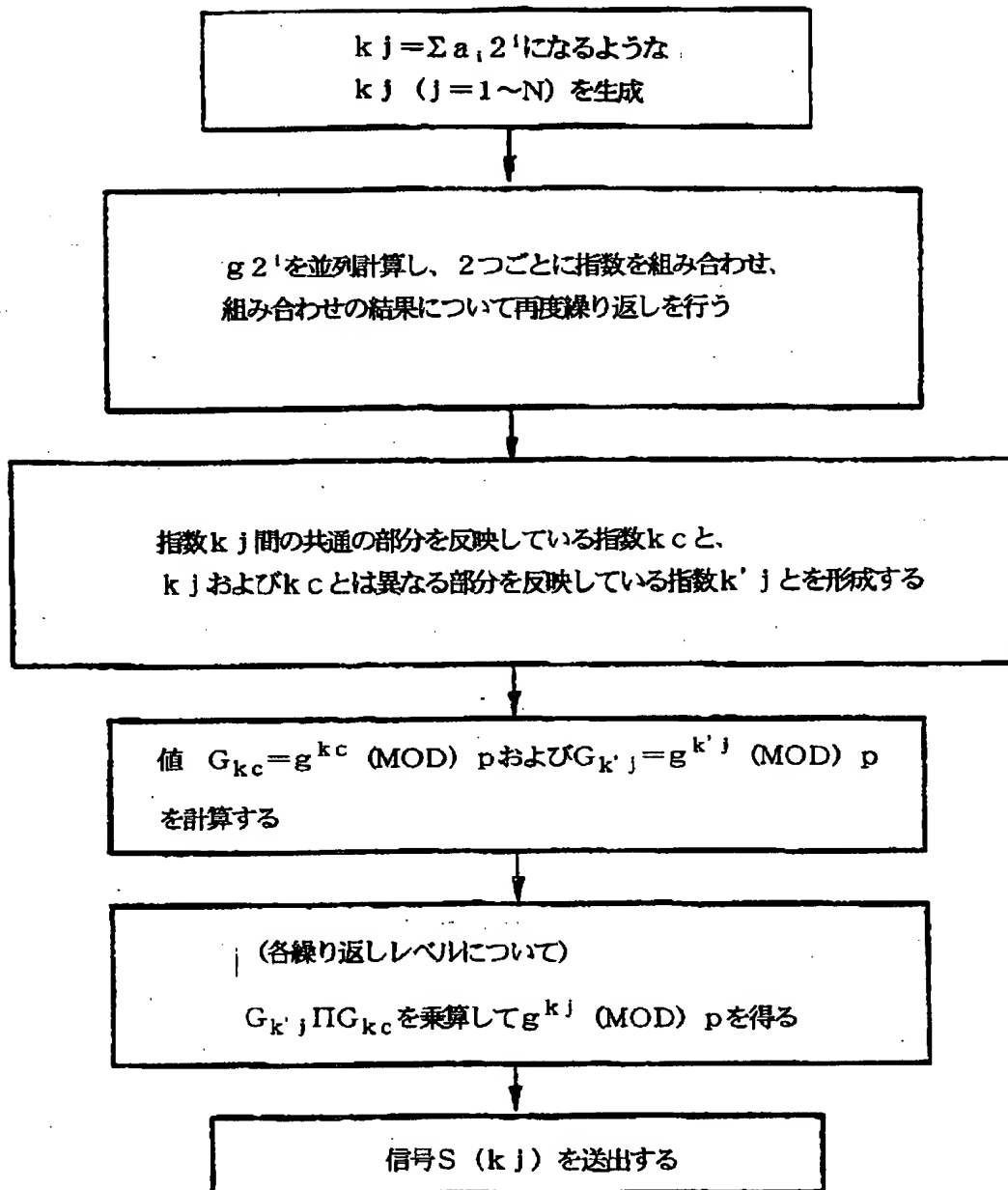


FIG.3

【図4】

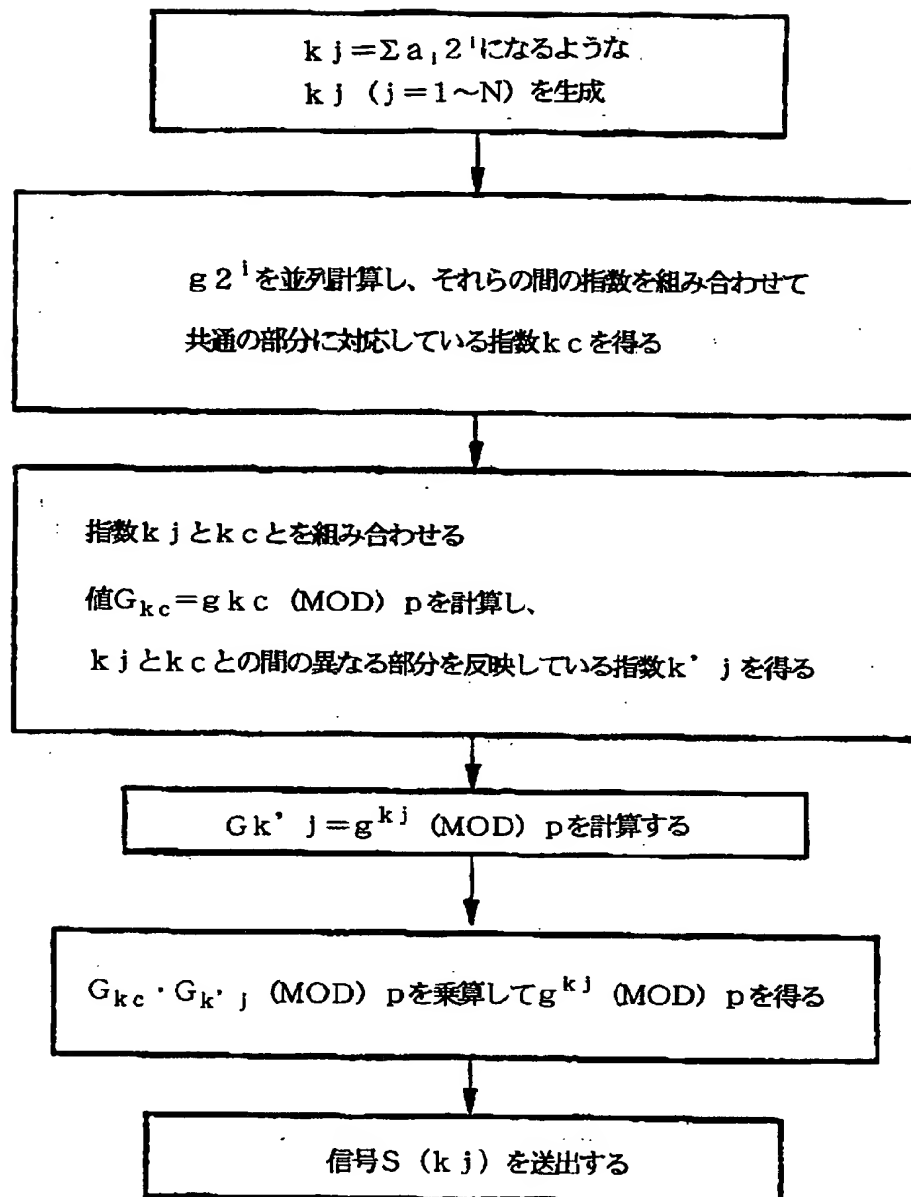


FIG.4

INTERNATIONAL SEARCH REPORT

Form PET/SA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Intern. Application No.

PCT/FR 96/00840

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>ELECTRONICS LETTERS, 17 AUG. 1989. STEVENAGE UK, vol. 25, no. 17, ISSN 0013-5194, pages 1171-1172, XP000054000 JEDWAB J ET AL: "Minimum weight modified signed-digit representations and fast exponentiation" see page 1171, left-hand column, last paragraph - right-hand column, line 17 see page 1171, right-hand column, line 41 - line 49</p>	1
A	<p>--- ADVANCES IN CRYPTOLOGY - AUSCRYPT '92. WORKSHOP ON THE THEORY AND APPLICATION OF CRYPTOGRAPHIC TECHNIQUES PROCEEDINGS, GOLD COAST, QLD., AUSTRALIA, 13-16 DEC. 1992. ISBN 3-540-57220-1, 1993, BERLIN, GERMANY, SPRINGER-VERLAG, GERMANY, pages 447-456, XP000470453 SUNG-MING YEN ET AL: "The fast cascade exponentiation algorithm and its applications on cryptography" see page 448, line 1 - page 449, line 4 -----</p>	3